Stronger Lower Bounds and Randomness-Hardness Trade-Offs Using Associated Algebraic Complexity Classes

Stronger Lower Bounds and Randomness-Hardness Trade-Offs Using Associated Algebraic Complexity Classes

Maurice Jansen and Rahul Santhanam

School of Informatics, The University of Edinburgh
Informatics Forum, 10 Crichton Street
Edinburgh, EH8 9AB, United Kingdom
`maurice.julien.jansen@gmail.com, rsanthan@inf.ed.ac.uk`

Abstract

We associate to each Boolean language complexity class $\mathcal{C}$ the algebraic class a·$\mathcal{C}$ consisting of families of polynomials $\{f_n\}$ for which the evaluation problem over $\mathbb{Z}$ is in $\mathcal{C}$. We prove the following lower bound and randomness-to-hardness results:

1. If polynomial identity testing (PIT) is in NSUBEXP then a·NEXP does not have *poly* size constant-free arithmetic circuits.
2. a·NEXP$^{\mathrm{RP}}$ does not have *poly* size constant-free arithmetic circuits.
3. For every fixed k, a·MA does not have arithmetic circuits of size n^k.

Items 1 and 2 strengthen two results due to Kabanets and Impagliazzo [7]. The third item improves a lower bound due to Santhanam [11].

We consider the special case low-PIT of identity testing for (constant-free) arithmetic circuits with low formal degree, and give improved hardness-to-randomness trade-offs that apply to this case.

Combining our results for both directions of the hardness-randomness connection, we demonstrate a case where derandomization of PIT and proving lower bounds are *equivalent*. Namely, we show that low-PIT $\in$ i.o-NTIME$[2^{n^{o(1)}}]/n^{o(1)}$ if and only if there exists a family of multilinear polynomials in a·NE/*lin* that requires constant-free arithmetic circuits of super-polynomial size and formal degree.

1998 ACM Subject Classification F.1.3 Complexity Measures and Classes.

Keywords and phrases Computational Complexity, Circuit Lower Bounds, Polynomial Identity Testing, Derandomization.

Digital Object Identifier 10.4230/LIPIcs.STACS.2012.519

1 Introduction

In this paper we study the arithmetic circuit complexity of families of multivariate polynomials $\{f_n\}$ in terms of the computational hardness of the underlying evaluation problem. Towards this end we associate to each Boolean language complexity class $\mathcal{C}$ the class a·$\mathcal{C}$ consisting of all families of polynomials $\{f_n\}$ with integer coefficients, such that given an integer input tuple x to f_n, an integer i and a bit b, it can be decided within the resources of the class $\mathcal{C}$ whether the ith bit of $f_n(x)$ equals b. We restrict the number of variables, the degree, and the bit size of coefficients of such families to be polynomially bounded in n (See Section 2 for the formal definition). We note that a similar notion was suggested by Koiran and Perifel [9].

29th Symposium on Theoretical Aspects of Computer Science (STACS'12).
Editors: Christoph Dürr, Thomas Wilke; pp. 519–530

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany


One of our main motivations is to find an elegant way to state "hybrid" results involving Boolean and arithmetic circuit lower bounds, such as the trade-offs of Kabanets and Impagliazzo [7] or the lower bound of Santhanam [11]. These are examples where people, perhaps unknowingly, have been proving lower bounds and randomness-hardness tradeoffs geared towards associated algebraic classes, while in our opinion lacking the proper language to describe, and consequently interpret, the results. The a·$\mathcal{C}$ notions provides a language for succinctly expressing these results, and leads to natural questions for making improvements. Consequently we have strengthened several important results from the literature.

A prime example of the above situation is the celebrated theorem by Kabanets and Impagliazzo [7], which says that if polynomial identity testing (PIT) is in NSUBEXP, then either NEXP $\not\subseteq$ P/*poly*, or permanent does not have poly-size arithmetic circuits. PIT is the problem of deciding for a given arithmetic circuit Φ whether it computes the zero polynomial. We refer to a recent survey by Saxena [12] for more on this problem. The quoted theorem tells us that derandomization of polynomial identity testing yields lower bounds of *some* sort. However, it doesn't tell us whether these will be Boolean lower bounds or arithmetic lower bounds. We make the observation[1] that the theorem by Kabanets and Impagliazzo is equivalent to the statement PIT $\in$ NSUBEXP $\Rightarrow$ a·NEXP/*lin* $\not\subseteq$ ASIZE$'$(*poly*). Here ASIZE$'$(*poly*) denotes the class of polynomial families $\{f_n\}$ computable by constant-free arithmetic circuit of size *poly*(n) using addition, multiplication, and division by computed constants (See Section 2). Hence, to answer the above question, putting PIT in NSUBEXP gives *arithmetic* lower bounds for families of polynomials that can be evaluated in NEXP with linear advice. A natural improvement to the result would be to drop the linear advice. We show that this can indeed be done[2], resulting in the following stronger theorem:

▶ **Theorem 1.** PIT $\in$ NSUBEXP $\Rightarrow$ a·NEXP $\not\subseteq$ ASIZE$'$(*poly*).

In the above, on the right hand side, the associated algebraic class gives us a measure of the explicitness of the lower bound. We have improved this explicitness from evaluable in NEXP/*lin* down to evaluable in NEXP. As it is generally undesirable to have non-uniform dependencies appearing in the explicitness measure of a lower bound, the main significance of our result is that we have managed to remove the non-uniformity.

Similar to Theorem 1 we observe that Theorem 5.2 of Ref. [7], which states that either NEXP$^{\mathrm{RP}}$ $\not\subset$ P/*poly* or Permanent does not have poly-size arithmetic circuits, is equivalent to the statement that a·NEXP$^{\mathrm{RP}}$/*lin* $\not\subseteq$ ASIZE$'$(*poly*). We also improve the explicitness of this lower bound and obtain that

▶ **Theorem 2.** a·NEXP$^{\mathrm{RP}}$ $\not\subseteq$ ASIZE$'$(*poly*).

Furthermore, we improve a theorem by Santhanam [11] which states that for every k, either MA $\not\subseteq$ SIZE(n^k), or there exists a family polynomial $\{f_n\}$, whose graph is decidable in MA, that is not in ASIZE(n^k). We show the following stronger result:

▶ **Theorem 3.** *For every k,* a·MA $\not\subseteq$ ASIZE(n^k).

The above results demonstrate the usefulness of the a·$\mathcal{C}$ notion. There are further reasons why the notion is worth exploring. It gives a way of bringing uniformity into the algebraic complexity setting. Note that traditional algebraic complexity classes such as VP and VNP

[1] A proof will appear in the full version of this paper.

[2] An obvious way to do this would be to 'just' show that a·NEXP $\subseteq$ ASIZE$'$(*poly*) $\Rightarrow$ a·NEXP/*lin* $\subseteq$ ASIZE$'$(*poly*). It is not clear whether this is true.

are inherently non-uniform. We also feel the notion could facilitate more interactions of techniques from structural complexity and algebraic complexity. Given how few lower bound techniques we have available, and given the well-known barriers such as natural proofs and algebrization to finding new ones, we need to make the best use of the ones we have. The recent lower bounds success of Williams [17] is an instructive example of how known techniques from different domains can be combined to give an interesting new result.

In general, one might ask for any known separation $\mathcal{C} \not\subseteq \mathcal{D}$ in the Boolean world whether it can be strengthened to show that a·$\mathcal{C} \not\subseteq$ a·$\mathcal{D}$. Note that this would indeed be a strengthening as $\mathcal{C} \subseteq \mathcal{D}$ trivially implies a·$\mathcal{C} \subseteq$ a·$\mathcal{D}$. Arithmetic analogues of time hierarchy results, eg., a·DTIME$[n^2] \subsetneq$ a·DTIME$[n^3]$ and a·NTIME$[n^2] \subsetneq$ a·DTIME$[n^3]$ can be proved quite easily using the fact that the separation can be witnessed by a unary language. However, we don't know whether arithmetic analogues of results such as Williams' lower bound hold. There could be a connection between proving the arithmetic analogue of a Boolean result and whether the techniques used to prove the Boolean result algebrize in the sense of Aaronson and Wigderson [1]. We have not properly explored this yet.

One of the advantages of using associated algebraic classes is that this enables us to derive tighter hardness-randomness trade-offs. This is especially striking for the case of the low-formal-degree polynomial identity testing problem (low-PIT). We define low-PIT as the special case of PIT for circuits Φ whose formal degree $deg(\Phi)$ is less than or equal to the size $|\Phi|$. Formal degree is a syntactic notion, easily computed for a circuit (See Section 2). Examples of types of circuits that automatically satisfy the degree restriction $deg(\Phi) \leq |\Phi|$ are formulas and skew-circuits, the latter being equivalent to algebraic branching programs. This makes low-PIT an important special case of the general problem. We show that we can specialize Theorem 1 to obtain the following low-degree version:

▶ **Theorem 4.** *low*-PIT $\in$ NSUBEXP $\Rightarrow$ a·NEXP $\not\subseteq$ ASIZEDEG$'(poly)$.

In the above ASIZEDEG$'(poly)$ is the class of families of polynomials $\{f_n\}$ computable by constant-free arithmetic circuits of size $poly(n)$ and formal degree $poly(n)$ (The 'prime' indicates that we allow a single division by a previously computed constant at the output gate).

For the special case of low-PIT, we also make progress on trade-offs that go in the opposite direction. Namely, we show that derandomization can be achieved under weaker hardness assumptions than was known previously. For example using our techniques we can prove the following theorem:

▶ **Theorem 5** (Proof to appear in full version)**.** *Suppose there exists a family* $\{p_n\} \in$ ml·NEXP *with* $\{p_n\} \notin$ i.o-ASIZEDEG$'(n^{e(n)})$*, where* $e(n)$ *is a monotone non-decreasing time constructible function with* $e(n) = \omega(1)$*. Then low*-PIT $\in$ NTIME$[2^{n^{o(1)}}]$.

In the above, ml·NEXP is the subclass of a·NEXP consisting of all families $\{f_n\}$, where each f_n is multilinear. The key improvement[3] that we make here over the techniques of Ref. [7], is that we can work with ASIZEDEG$'$-hardness instead of ASIZE$'$-hardness in case we only need to cater for low-PIT. To achieve such improved trade-offs, we prove a so-called root extraction lemma (Lemma 19) that is formal-degree efficient. This lemma, which is of independent interest, is subsequently combined with the framework of Ref. [7]. As an additional twist, we start with a hardness assumption in terms of an associated algebraic class.

[3] See some remarks about the difference in Section 2.

Finally, combining our results for both directions of the hardness vs. randomness connection, the work of this paper culminates with the following theorem, which demonstrates a setting where derandomization of PIT and proving lower bounds are *formally equivalent*:

▶ **Theorem 6.** *There exists a family* $\{p_n\} \in \text{ml·NE}/lin$ *with* $\{p_n\} \notin \text{ASIZEDEG}'(s(n))$, *for* $s(n) = n^{\omega(1)}$ *if and only if low-*PIT $\in \text{i.o-NTIME}[2^{r(n)}]/r(n)$, *for* $r(n) = n^{o(1)}$.

In the past there have been several authors claiming partial converses to randomness-hardness theorems involving PIT. Our paper is the first where an actual equivalence is being observed. In this the associated algebraic classes play a central role, which we offer as further evidence of the importance of this notion.

2 Preliminaries

Let $\text{NSUBEXP} = \cap_\epsilon \text{NTIME}[2^{n^\epsilon}]$. We define $\text{SIZE}(s(n))$ to be the class of all languages in $\{0,1\}^*$ computable by Boolean circuits of size $s(n)$. A (division-free) arithmetic circuit over some field $\mathbb{F}$ and a set of variables $X = \{x_1, x_2, \ldots, x_n\}$ is given by a labeled directed acyclic graph. Nodes of in-degree zero are labeled with elements of $X \cup \mathbb{F}$. Other nodes are labeled by $+$ or $\times$. To each node, a.k.a. gate, we can associate a polynomial $\in \mathbb{F}[X]$, defined inductively in the obvious way. If constant-labels are restricted to be in $\{-1, 0, 1\}$ the circuit is called constant-free. For the size of an arithmetic circuit we count the number of wires. We define $\text{ASIZE}(s(n))$ to be the class of all families of polynomials $\{f_n\}$ with integer coefficients that have constant-free arithmetic circuits of size $s(n)$. We let $\text{ASIZE}'(s(n))$ be the class obtained from $\text{ASIZE}(s(n)$ by allowing one single division at the output gate by an integer $a \neq 0$, where a has been computed by the circuit. We remark that due to a result by Strassen [14] on avoidance of divisions, cf. Theorem 2.17 and Corollary 3.9 in [7], a family of polynomials $\{f_n\}$ of $poly(n)$ degree can be computed by an arithmetic circuit of $poly(n)$ size *with arbitrary use of division gates* iff $\{f_n\} \in \text{ASIZE}'(poly(n))$. We define "infinitely often" versions of these classes in the obvious way. For example $\text{i.o-ASIZE}(s(n))$ is the class of families $\{f_n\}$ such that for infinitely many n, f_n can be computed by a size $s(n)$ circuit.

$\text{ASIZEDEG}(s(n))$ is obtained from $\text{ASIZE}(s(n))$ by adding the restriction that formal degree of the circuit is bounded by $s(n)$ as well. Formal degree is defined inductively as follows. For input gates, regardless of their label, formal degree is 1. Formal degree of an addition gate is taken to be the maximum of the formal degree of its inputs. For multiplication gates one takes the sum of formal degrees of its inputs. We define the class $\text{ASIZEDEG}'(poly)$ to be the class of families of polynomials $\{f_n\}$ with integer coefficients such that there exist families $\{g_n\}$ and $\{c_n \in \mathbb{Z}\}$ in $\text{ASIZEDEG}(poly)$ with $f_n = g_n/c_n$, for each n.

Families in $\text{ASIZE}(poly)$ can have super-polynomial degree, e.g. x^{2^n} can be computed with $n-1$ repeated multiplications. We like to point out that in general for a family $\{f_n\} \in \text{ASIZE}(poly)$ with $deg(f_n) = n^{O(1)}$ *it is not known* whether $\{f_n\} \in \text{ASIZEDEG}(poly)$. In particular it is a fallacy to think the well-known trick of computing degree components separately at every gate in the circuit proves this, as was pointed[4] out by Bürgisser [3], cf. [8]. Namely, this construction requires a model where arbitrary constants can be used by the circuit at unit cost. A similar remark can be made for the classes $\text{ASIZEDEG}'(poly)$

[4]The class $\text{ASIZEDEG}(poly)$ is known as VP^0 in the literature, whereas $\text{ASIZE}(poly)$ corresponds to families of polynomials with τ-complexity $poly(n)$, cf. Ref. [9].

and ASIZE′(*poly*), in which case it is perhaps more obvious that computing components separately does not help, since one of the given circuits only computes a constant.

We define the language corresponding to the polynomial identity testing problem PIT = $\{\Phi : \Phi$ is a division-free constant-free arithmetic circuit such that $\Phi \equiv 0\}$. Similarly, we define low-PIT to be the following language

$\{\Phi : \Phi$ is a division-free constant-free arithmetic circuit of formal degree $\leq |\Phi|$ and $\Phi \equiv 0\}$.

We make use of the well-known Schwartz-Zippel-deMillo-Lipton Lemma.

▶ **Lemma 7** ([4, 13, 18]). *Let A be an arbitrary nonempty subset of the field $\mathbb{F}$. Then for any nonzero polynomial $f \in \mathbb{F}[X]$ of degree d, $\Pr[f(a_1, a_2, \ldots, a_n) = 0] \leq \frac{d}{|A|}$, where the a_i's are picked independently and uniformly at random from A.*

We use several easily proved propositions.

▶ Proposition 1. A constant-free division-free arithmetic circuit of size s and formal degree d without variables computes an integer constant of absolute value at most 2^{ds}.

By hard-wiring inputs we obtain the following corollary:

▶ **Corollary 8.** *For a constant-free division-free arithmetic circuit of size s and formal degree d computing a polynomial $f(x_1, x_2, \ldots, x_n)$, if we evaluate f on integers $a_1, \ldots, a_n$ of at most B bits, then $|f(a_1, a_2, \ldots, a_n)| \leq 2^{O(dsB^2)}$.*

▶ Proposition 2 (Multilinear Extension over $\mathbb{Z}$). Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function. Define $F(x_1, \ldots, x_n) = \sum_{a \in \{0,1\}^n} f(a) \prod_{i \in [n]} (1 - x_i + a_i(2x_i - 1))$. Then F is a multilinear polynomial with integer coefficients that coincides with f on $\{0,1\}^n$. Furthermore, F is the unique polynomial with these properties.

We now get to the central definition of this paper.

▶ **Definition 9.** Let $\mathcal{C}$ be a language complexity class. Corresponding to $\mathcal{C}$ we have the associated algebraic class a·$\mathcal{C}$ which is given by the collection of all polynomial families $\{f_n\}$ defined in $m(n) = n^{O(1)}$ variables of degree $poly(n)$ having integer coefficients of $poly(n)$ bit size such that the evaluation language

$$E(\{f_n\}) := \{(1^n, a_1, a_2, \ldots, a_{m(n)}, i, b) : \text{the } i\text{th bit of } f_n(a_1, a_2, \ldots, a_{m(n)}) \text{ equals } b\} \in \mathcal{C},$$

where i, $a_1, a_2, \ldots, a_{m(n)} \in \mathbb{Z}$ are given in binary. We denote the subclass of a·$\mathcal{C}$ consisting of families of multilinear polynomials by ml·$\mathcal{C}$.

Typically for a complexity class $\mathcal{C}$ we will have the complementation property that a·$\mathcal{C}$ = a·($\mathcal{C} \cap$ co$\mathcal{C}$). This is due to the inclusion of the bit b in the definition of the evaluation language. We have the following property in particular:

▶ Proposition 3. a·NEXP = a·(NEXP ∩ coNEXP).

Namely, given a NEXP-machine M deciding $E(\{f_n\})$ for some family of polynomials $\{f_n\}$, one can simulate M on inputs $(1^n, a_1, a_2, \ldots, a_{m(n)}, i, b)$ and $(1^n, a_1, a_2, \ldots, a_{m(n)}, i, 1-b)$. In these nondeterministic simulations one finds at most one of them accepting, and it is guaranteed there exists at least one such path. For all paths where an accept is found, the machine knows exactly whether $(1^n, a_1, a_2, \ldots, a_{m(n)}, i, b) \in E(\{f_n\})$. This means that we have a nondeterministic exponential time *flag machine* for computing the characteristic function of $E(\{f_n\})$, which implies that $E(\{f_n\}) \in$ NEXP ∩ coNEXP. Recall that a flag machine sets a flag bit and produces output. If the flag is 0 this means on the given path no output is produced. If the flag is 1 it signals the output is valid. To compute a function, all flag = 1 paths must produce the same value, and there must be at least one such path.

3 Improved Lower Bounds from Derandomization of PIT

In this section, in order to avoid ambiguity we use a new variable N to indicate the input length for Boolean complexity classes. For example, $\Sigma_4\text{TIME}[N]$ is the class of all languages decidable by Σ_4-machines in time $O(N)$ for inputs of size N. We will prove Theorem 2, and the following theorem (which implies Theorem 1):

▶ **Theorem 10.** PIT $\in$ NSUBEXP $\Rightarrow$ ml·NEXP $\not\subseteq$ ASIZE$'(poly)$.

We first establish fixed polynomial size arithmetic circuit lower bounds for a·PH.

▶ **Theorem 11.** *For any fixed k, there exists $\{f_n\} \in$ a·PH with $\{f_n\} \notin$ i.o-ASIZE$'(n^k)$. Furthermore, each f_n is multilinear in n variables, has degree $3k$ and has coefficients in $\{0,1\}$.*

Proof. For simplicity we show a fixed size lower bound in terms of ASIZE instead of ASIZE$'$. The proof is easily modified to yield the more general statement. There are $2^{O(n^{2k})}$ arithmetic circuits of size at most n^k. Consider the class $\mathcal{F}$ of homogeneous multilinear polynomials in n variables of degree $3k$ with $0,1$ coefficients. Then $|\mathcal{F}| = 2^{\binom{n}{3k}}$. Hence there exists $f_n \in \mathcal{F}$ that is not in ASIZE(n^k). Our goal is to find it 'in PH'.

Let $\mathcal{C}$ be the class of arithmetic circuits corresponding to $\mathcal{F}$, where we just represent in the $\Sigma\Pi$-form, i.e. a sum of monomials. We can fix some representation of $\mathcal{C}$ by strings of length $O(n^{3k})$. Our goals is to find the lexicographically least circuit $\Phi \in \mathcal{C}$ such that *for all* arithmetic circuits Ψ of size n^k, $\Phi - \Psi \not\equiv 0$. Define the language L to consist of tuples $(1^n, < \Phi >)$ with the property that for all circuits Ψ of size n^k, $\Phi - \Psi \not\equiv 0$, where $< \Phi >$ to denotes the string encoding of Φ. Checking $\Phi - \Psi \not\equiv 0$ is a coRP predicate. This implies that L is in coNP^{RP}. On input 1^n, using binary search and making *existential* queries to L, one can find the lexicographically least Φ of size $O(n^{3k})$ such that $(1^n, < \Phi >) \in L$ in $\text{FP}^{\text{NP}^{\text{coRP}^{\text{RP}}}}$. Define f_n to be the polynomial computed by this Φ. Once the sum of monomials representations of f_n is known, evaluations is *poly*-time computable for integer inputs. Hence we obtain that $E(\{f_n\}) \in$ PH. ◀

From the proof of Theorem 11 we can conclude that the following lemma is true:

▶ **Lemma 12.** *There exists a constant $c_1 \in \mathbb{N}$, such that for any $k \geq 1$, there exists $\{f_n\} \in$ ml· $\Sigma_4\text{TIME}[N^{c_1 k}]$ with $\{f_n\} \notin$ i.o-ASIZE$'(n^k)$.*

Namely, to describe an algorithm for $E(\{f_n\})$, consider an input $(1^n, a_1, \ldots, a_n, i, b)$ of size N. The proof of Theorem 11 shows that we can first find in $\Sigma_4\text{TIME}[poly(n^{3k})]$ a sum-of-monomials description of a polynomial f_n of size $O(n^{3k})$ that requires size n^k. After that we evaluate $f(a_1, \ldots, a_n)$, which can be done in time $poly(N^{3k})$ given this simple representation of f_n. We get that the total overhead for deciding $E(\{f_n\})$ is $\Sigma_4\text{TIME}[poly(N^k)]$. One now easily derives the following lemma:

▶ **Lemma 13.** *There exists a constant $c_2 \in \mathbb{N}$, such that for any $k \geq 1$, there exists $\{f_n\} \in$ ml·$\text{DTIME}^{0,1\text{-}Perm^{[1]}}[N^{c_2 k}]$ with $\{f_n\} \notin$ i.o-ASIZE$'(n^k)$.*

Proof. By Toda's theorem [15] and Valiant's Completeness result [16], we know that there exists an absolute constant $b \in \mathbb{N}$ so that for every $k \in \mathbb{N}$, $\Sigma_4\text{TIME}[N^k] \subseteq \text{DTIME}^{0,1\text{-Perm}^{[1]}}[N^{bk}]$. Let c_1 be the constant given by Lemma 12. We get that $\Sigma_4\text{TIME}[N^{c_1 k}] \subseteq \text{DTIME}^{0,1\text{-Perm}^{[1]}}[N^{bc_1 k}]$. Hence the lemma holds for $c_2 = bc_1$. ◀

We use the following lemma by Kinne, van Melkebeek and Shaltiel:

▶ **Lemma 14** (Claim 5 in [5]). *There exists a constant d such that the following holds for any functions $a(\cdot)$ and $t(\cdot)$ with $a(\cdot)$ time-constructible and $t(\cdot)$ monotone. If* PIT $\in$ NTIME$(t(N))$ *and* $\{per_n\} \in$ ASIZE$'(a(n))$, *then* DTIME$^{0,1\text{-}Perm[1]}[N] \subseteq$ NTIME$[t(N \cdot \log^d N \cdot a(\sqrt{N}))]$.

We are now ready to prove Theorem 10.

Proof. *(Theorem 10)* We are done if $\{per_n\} \notin$ ASIZE$'(poly)$, so assume that ASIZE$'(\{per_n\}) \leq n^\ell$, for $\ell \in \mathbb{N}$. Consider arbitrary $k \geq 1$. Combining Lemmas 13 and 14, we obtain that for any monotone function $t(\cdot)$, if PIT $\in$ NTIME$[t(N)]$, then ml·NTIME$[t(N^{c_2 k} \cdot \log^d N^{c_2 k} \cdot N^{c_2 \ell k/2})] \not\subseteq$ ASIZE$'(n^k)$. As we are assuming that PIT $\in$ NSUBEXP, if we apply this with $t(N) = 2^{N^\epsilon}$, for small enough ϵ, we get that ml·NTIME$[2^N] \not\subseteq$ ASIZE$'(n^k)$. Since k was arbitrary, we get that ml·NTIME$[2^N] \not\subseteq$ ASIZE$'(poly)$, which implies that ml·NEXP $\not\subseteq$ ASIZE$'(poly)$. ◀

Next we move on to the proof of Theorem 2.

Proof. *(Theorem 2).* Suppose that a·NEXP$^{\text{RP}} \subseteq$ ASIZE$'(poly)$. Then a·EXP $\subseteq$ ASIZE$'(poly)$. We claim that this implies that EXP $\subseteq$ SIZE$(poly)$. Let $L \in$ EXP be any language. We will show that $L \in$ SIZE$(poly)$. Since we can evaluate multilinear extensions (Proposition 2) of characteristic functions of EXP languages within EXP itself, we get $\{F_n\}$ in a·EXP, where F_n is the multilinear extension of χ_L on $\{0,1\}^n$. We get that $\{F_n\} \in$ ASIZE$'(poly)$. This means that we have constant-free (division-free) arithmetic circuits Φ_1 and Φ_2 of size at most $p(n) = n^{O(1)}$, such that Φ_2 does not contain variables and computes some nonzero constant $c \in \mathbb{Z}$. Furthermore, if Φ_1 computes G_n then it holds that $G_n = c \cdot F_n$. For input $a \in \{0,1\}^n$, $F_n(a) \in \{0,1\}$, which means for such inputs $G_n(a) \in \{0, c\}$. We want to evaluate Φ_1 modulo some prime number q that does not divide c. This will tell us $\chi_L(a)$. We have that $|c| \leq 2^{2^{p(n)}}$ due to Proposition 1. This means that c has at most $2^{p(n)}$ prime factors. Hence, using the Prime Number Theorem there exists a prime number q of $p(n)^2$ bits, provided n is large enough, that does not divide c. As our task is to show only the non-uniform upper bound $L \in$ SIZE$(poly)$, mere existence of this number q suffices for our purposes, as we can hardcode it into the Boolean circuit simulating Φ_1 and Φ_2. Hence EXP $\subseteq$ SIZE$(poly)$.

Babai, Fortnow, Lund [2] prove that EXP $\subseteq$ SIZE$(poly) \Rightarrow$ EXP = MA. So we get that EXP = MA. Also, because easily $\{per_n\} \in$ a·NEXP$^{\text{RP}}$, we have that $\{per_n\} \in$ ASIZE$'(poly)$. This implies that P$^{\#\text{P}} \subseteq$ NP$^{\text{RP}}$, cf. Lemma 5.3 in Ref. [7]. By Toda's Theorem [15], MA $\subseteq$ P$^{\#\text{P}}$. Hence we obtain that EXP = MA $\subseteq$ NP$^{\text{RP}}$. By padding this implies that EEXP $\subseteq$ NEXP$^{\text{RP}}$. Hence a·EEXP $\subseteq$ a·NEXP$^{\text{RP}} \subseteq$ ASIZE$'(poly)$. This is a contradiction. One can easily deduce that a·EEXP $\not\subseteq$ i.o-ASIZE$'(n^{\log n})$ by observing that Lemma 12 also holds if we allow k to depend on n as $k(n) = \lceil \log n \rceil$. ◀

We can specialize Lemma 14 so that we replace the condition "PIT $\in$ NTIME$(t(N))$ and $\{per_n\} \in$ ASIZE$'(a(n))$" by "low-PIT $\in$ NTIME$(t(N))$ and $\{per_n\} \in$ ASIZEDEG$'(a(n))$". This yields the following theorem (which implies Theorem 4):

▶ **Theorem 15.** *low*-PIT $\in$ NSUBEXP $\Rightarrow$ ml·NEXP $\not\subseteq$ ASIZEDEG$'(poly)$.

4 Stronger Fixed Size Lower Bounds for MA

As the result we aim to strengthen puts somewhat different constraints on constants appearing in arithmetic circuits compared to what we have seen so far, we make the following provisional definition. Let $\text{ASIZE}^{\text{free}}(s(n))$ denote the class obtained from $\text{ASIZE}(s(n))$ by granting the underlying circuits arbitrary constant labels $\in \mathbb{Z}$. Similar to Theorem 11 we have the following theorem.

▶ **Theorem 16** (Proof will appear in full version). *$\forall k \exists \{f_n\} \in$ a·PH with $\{f_n\} \notin$ i.o-$\text{ASIZE}^{free}(n^k)$.*

We want to strengthen Theorem 1.4 of [11], which we can reformulate it in our terminology as follows:

▶ **Theorem 17** ([11]). *For every k, either 1) $\text{MA} \not\subseteq \text{SIZE}(n^k)$, or 2) $\text{a·MA} \not\subseteq \text{ASIZE}^{free}(n^k)$.*

We will show that for every k, the second item holds by itself. Let us briefly remark on a technical issue related to this reformulation. For $\{f_n\}$, where f_n is a integer polynomial over n variables, Ref. [11] uses the notion $Gh(\{f_n\}) = \{(\vec{x}, v) | f_n(\vec{x}) = v\}$, and proves that for every k, either $\text{MA} \notin \text{SIZE}(n^k)$, or there exists $\{f_n\} \notin \text{ASIZE}^{\text{free}}(n^k)$ with $Gh(\{f_n\}) \in \text{MA}$. We prefer to work with the evaluation language $E(\{f_n\})$ instead of $Gh(\{f_n\})$. One can observe that the argument we give to strengthen Theorem 17 can be easily modified to work with $Gh(\cdot)$ instead. Consider the following proposition:

▶ Proposition 4. If $\{per_n\} \in \text{ASIZE}^{\text{free}}(poly)$, then 1) 0, 1-permanent of an $n \times n$ matrix over $\mathbb{Z}$ can be computed with $poly(n)$ size Boolean circuits, and 2) $\text{PH} \subseteq \text{MA}$.

For the above, it is argued in Ref. [11], proof of Theorem 1.4, that the first item follows from $\{per_n\} \in \text{ASIZE}^{\text{free}}(poly)$, and that the second item follows from the first. The following theorem implies Theorem 3 from the introduction:

▶ **Theorem 18.** *For any fixed k, there exists $\{f_n\} \in \text{a·MA}/\text{ASIZE}^{free}(n^k)$.*

Proof. We show that Item 2 of Theorem 17 holds by itself. For this, we indicate how the proof of Theorem 1.4 in Ref. [11] must be modified. This proof conditions on the predicate $\{\text{per}_n\} \in \text{ASIZE}^{\text{free}}(poly)$. If this is not true, the proof there can easily be modified to use $E(\cdot)$ instead of $Gh(\cdot)$, which then yields the statement of the theorem. Otherwise, suppose that $\{\text{per}_n\} \in \text{ASIZE}^{\text{free}}(poly)$. By Proposition 4 we have that $\text{PH} \subseteq \text{MA}$. The latter implies that $\text{a·PH} \subseteq \text{a·MA}$. Hence in this case Item 2 holds also, due to Theorem 16. ◀

5 A Characterization of Derandomization for low-PIT

We will use the algebraic hardness-to-randomness framework of Ref. [7]. The refinement that we make here is to show that it suffices to start with a weaker[5] $\text{ASIZEDEG}'$-hardness assumption rather than ASIZE'-hardness, in case we only need to cater for low-PIT.

For a polynomial $f(x, y) \in \mathbb{F}[x_1, \ldots, x_n, y]$ and $p(x) \in \mathbb{F}[x_1, \ldots x_n]$, $f_{|y=p}$ denotes the polynomial obtained by substituting p for y in f. We will also write this polynomial as $f(x, p)$. In case $f_{|y=p} = 0$, we say that p is a *root* of f for y. The following is our degree-efficient root extraction lemma:

[5] See Section 2 for some remarks pertaining to these measures when dealing with families of *poly*-degree.

▶ **Lemma 19.** *Suppose that $f \in \mathbb{Z}[x_1, \ldots, x_n, y]$ is a nonzero polynomial computed by a division-free constant free arithmetic circuit of size s and formal degree D. Suppose that $p \in \mathbb{Z}[x_1, \ldots, x_n]$ is a root of f for y. Then there exist constant-free division-free arithmetic circuits Φ_1 and Φ_2 of size and formal degree bounded by $poly(n, s, D, L)$ such that the following are true:*

1. *Φ_1 computes a polynomial $q \in \mathbb{Z}[x_1, \ldots, x_n]$.*
2. *Φ_2 does not contain variables. It computes a nonzero constant $c \in \mathbb{Z}$.*
3. *It holds that $c \cdot p = q$.*
4. *L bounds the maximum bit size of $p(x)$ on $\{0, 1, \ldots, d_p d_f\}^n$, where d_f and d_p are the degrees of f and p, respectively.*

The proof of the above lemma follows by analyzing the degree blow-up in the root extraction method of Ref. [6]. As this procedure involves Newton iteration it is a priori not at all clear that formal-degrees are well-behaved, but this turns out to be true. The proof will appear in the full version of this paper. We continue towards our hardness-to-randomness trade-offs. First we need the following lemma:

▶ **Lemma 20** (Nisan-Wigderson Design [10])**.** *Let n, m be integers with $n < 2^m$. There exists a family of sets $S_1, S_2, \ldots, S_n \subseteq [\ell]$, such that 1) $\ell = O(m^2/\log n)$, 2) For each i, $|S_i| = m$, and 3) For every $i \neq j$, $|S_i \cap S_j| \leq \log n$. Furthermore, the above family of sets can be computed deterministically in time $poly(n, 2^\ell)$.*

Define NW^p as follows. For parameters ℓ, m, n, construct the set system $S_1, S_2, \ldots, S_n$ as in Lemma 20. Then for $a_1, a_2, \ldots, a_\ell \in \mathbb{F}$, and a polynomial p in m variables, $NW^p(a) = (p(a_{|S_1}), p(a_{|S_2}), \ldots, p(a_{|S_n}))$. The following lemma is derived from Lemma 19 using a hybrid argument, cf. Lemma 7.6 in [7]:

▶ **Lemma 21.** *Let n and m be integers with $n < 2^m$ and $m < n$. Suppose we are given a nonzero polynomial $f \in \mathbb{Z}[y_1, \ldots, y_n]$ of degree d_f and a multilinear polynomial $p \in \mathbb{Z}[x_1, \ldots, x_m]$ with coefficients of bit size at most m^e, for some integer constant $e \geq 1$. Assume that f can be computed by a division free constant-free arithmetic circuit of size s and formal degree D. Let $S \subseteq \mathbb{Z}$ be any set of size $|S| > d_f m$, and let ℓ be given by Lemma 20. Suppose that $f(NW^p(a)) = 0$ for all $a \in S^\ell$. Then there exists $q \in \mathbb{Z}[x_1, \ldots, x_m]$ and $c \in \mathbb{Z}/\{0\}$ such that $p = q/c$, where q and c can be computed by constant-free division-free arithmetic circuits of size and formal degree $poly(n, m^e, s, D)$.*

A proof of the above lemma will be included in the full version of the paper. Our first trade-off is as follows:

▶ **Theorem 22.** ml·NEXP $\not\subseteq$ ASIZEDEG$'(poly(n)) \Rightarrow$ *low*-PIT $\in \bigcap_{\epsilon>0}$ i.o-NTIME$[2^{N^\epsilon}]$.

Proof. Consider a family $\{p_m\} \in$ ml·NEXP that is not in ASIZEDEG$'(poly)$. By reindexing we can assume wlog. that p_m is defined over m variables. Let e be such that coefficients of p_m are at most m^e bits. We have that for every k, there exist infinitely many m such that p_m cannot be written as $p_m = f_m/c_m$, where f_m and $c_m \in \mathbb{Z}/\{0\}$ are computed by constant-free division-free arithmetic circuits of size and formal degree at most m^k. The $m \in \mathbb{Z}$ that satisfy this property we call the *good indexes for* k. We use the fact that a·NEXP = a·(NEXP $\cap$ coNEXP). This means that we there exists a constant d and a nondeterministic flag machine M running in time $2^{(n')^d}$ for inputs of size n' that can compute the characteristic function of $E(\{p_m\})$ on a given input, cf. Proposition 3.

Let c_0 be an absolute constant that bounds the overhead of Lemma 21, in the sense that for the case $n = s = D$ we can write an upper bound of $n^{c_0} m^{ec_0}$ for the bound

$poly(n, m^e, s, D)$ given by the lemma. We will describe an i.o-NSUBEXP algorithm for low-PIT. Let Φ be a constant-free (division-free) arithmetic circuit of size N computing f. First we check that the formal degree of Φ is bounded by N, if not reject.

Let $m = \lfloor N^{1/r} \rfloor$, where r is chosen arbitrarily large. We claim that for infinitely many input lengths N the following test property holds: for every constant-free arithmetic circuit Ψ of size N, $\Psi \equiv 0 \Leftrightarrow (\forall a \in S^\ell), \Psi(NW^{p_m}(a)) = 0$, where $S = [Nm+1]$ with $\ell = O(m^2/\log N)$ taken according to Lemma 20. This follows from Lemma 21. Namely, let $k = c_0(r+e)$ and let $\mathcal{M}$ be the set of good indexes for k. Then $\mathcal{M}$ is an infinite set. Consider input lengths N of the form $N = (N')^r$, where $N' \in \mathcal{M}$. For such N, we set $m = N'$. The test property can only be violated if for some Ψ of size N we have that $\Psi \not\equiv 0$, while $(\forall a \in S^\ell), \Psi(NW^{p_m}(a)) = 0$. By Lemma 21 we obtain that p_m can be written as $p_m = f_m/c_m$, for f_m and $c_m \in \mathbb{Z}/\{0\}$ that are computed by constant-free arithmetic circuits of size and formal degree at most $N^{c_0} m^{c_0 e} = (m)^{c_0(r+e)} = m^k$. We know the latter does not hold for $m \in \mathcal{M}$.

We continue the description of the algorithm. We produce a set H to sample Φ with, namely we take H to be the output of $NW^{p_m}(\cdot)$ on S^ℓ. We have that $|H| = (Nm+1)^\ell = 2^{O(N^{2/r}\log N)}$. We do simulations of the machine M for $E(\{p_m\})$ to get all the bits of all the evaluations of $p_m(\cdot)$ on S^ℓ. As p_m is multilinear with coefficients of bit length at most m^e, we can bound the bit size of any such evaluation by $O(m^e \log(Nm))$. This means that the inputs $(1^m, a_1, \ldots, a_m, i, b)$ that we simulate M on, have bit size $O(m \log(Nm))$ (recall that i is given in binary). The simulation for a single such input thus costs $\text{NTIME}[2^{O(m^d \log^d(Nm))}] = \text{NTIME}[2^{O(N^{d/r}\log^d N)}]$. To get all bits of an evaluation for a single element in H therefore takes at most $\text{NTIME}[O(m^e \log(Nm)) \cdot 2^{O(N^{d/r}\log^d N)}]$, which we can bound as $\text{NTIME}[2^{O(N^{d/r}\log^d N)}]$. To construct the entire set H we can use the same asymptotic time bound assuming wlog. that $d \geq 2$.

If during the process of obtaining all the bits we obtain a flag bit set to 0, we reject. This means that on every path where we pass this check, we have obtained a hitting set, unless N is an input length where the test property is not satisfied. On these paths, we continue to verify deterministically that $f(h) = 0$ for all $h \in H$. If yes, then we accept, else reject. By our previous remarks, for infinitely many N, this correctly decides whether $\Phi \equiv 0$.

Let us consider the cost of evaluation of Φ on elements of H. For $a \in S^\ell$ and subset S_j in the Nisan-Wigderson design, the bit size of $p_m(a_{|S_j})$ is $O(m^e \log(Nm))$. By Corollary 8 this means that the absolute value of any gate of Φ for input $NW^{p_m}(a)$ is at most $2^{O(N^2 m^{2e} \log^2(Nm))} = 2^{N^{O(1)}}$. Thus intermediate values can be represented by $poly(N)$ bits. We conclude that evaluation of Φ on a single element of the test set H cost time $poly(N)$. We can conclude the entire cost of our test algorithm is $\text{NTIME}[2^{O(N^{d/r}\log^d N)}]$. As r can be chosen arbitrarily large and d is an absolute constant not depending on r, we conclude that low-PIT $\in \bigcap_{\epsilon>0}$ i.o-NTIME$[2^{N^\epsilon}]$. ◀

5.1 Proof of Theorem 6

We first prove the hardness-to-randomness direction. The following corollary to Lemma 21 follows straightforwardly:

▶ **Corollary 23.** *Let $s(n) = n^{\omega(1)}$ be a function. Suppose that $\{p_n\}$ is a family of multilinear polynomials in n variables with coefficients of bit size at most $n^{e'}$ for some integer e', such that p_n cannot be written as q_n/c_n for $c_n \in \mathbb{Z}\backslash\{0\}$ for any q_n and c_n computed by constant-free arithmetic circuits of size $s(n)$. Then there exists an absolute constant $c > 0$ such that for any division-free constant-free arithmetic circuit Φ of size n with $deg(\Phi) \leq n$, if we take*

m such that $s(m) \cdot m^{-e'c} > n^c$ and let ℓ be given by Lemma 20, then for all large enough n, $\Phi \equiv 0 \Leftrightarrow (\forall a \in S^\ell), \Phi(NW^{p_m}(a)) = 0$, where $S = [nm+1]$.

We will describe an i.o-NTIME$[2^{n^{o(1)}}]/n^{o(1)}$ algorithm for low-PIT. Let Φ be an arithmetic circuit of size N, and let f be the polynomial computed by it. First we check that the formal degree of Φ is bounded by N, if not reject. Else, consider the given family $\{p_m\}$. By reindexing we may assume wlog. that p_m is defined over m variables. Let $e' \geq 1$ be such that p_m has coefficients of bit size at most $m^{e'}$. We have that for infinitely many m, p_m has ASIZEDEG′-hardness larger than $s(m)$, where $s(m) = m^{\omega(1)}$. The m that have this property we call *good*.

We use the complementation property for ml·NE/*lin*, cf. Proposition 3 and the comment thereafter. This means that we have a nondeterministic flag machine M running in time $2^{O(n')}$ with $O(n')$ bits of advice for inputs of size n' that can compute the characteristic function of $E(\{p_m\})$ Let c be the constant given by Corollary 23. For input size N the algorithm receives two strings of advice α and β. First, if there exists a good m_0 such that $s(m_0)(m_0)^{-ce'} \in [N^c, (N+1)^c]$, then $\alpha = 1^{m_0}$. If there is no such m_0, then α is set to the empty string. A simple argument shows that $|\alpha| = N^{o(1)}$. For the second piece of advice β we obtain the advice M needs so we can complete the simulations which we describe below (we will analyze this in more detail there).

In case the algorithm receives the empty string for α, it halts and rejects. Otherwise, we set $m = m_0$. Note that as N^c is a strict monotone increasing function it must be that for infinitely many N we obtain a good m_0 as advice. By Corollary 23, provided N is large enough, the following test property holds: $\Phi \equiv 0 \Leftrightarrow (\forall a \in S^\ell), \Phi(NW^{p_m}(a)) = 0$, where $S = [Nm+1]$ with $\ell = O(m^2/\log N)$ taken according to Lemma 20.

Let us continue the description of the algorithm. We produce a set H to sample Φ with, namely take H to be the output of $NW^{p_m}(\cdot)$ on S^ℓ. We have that $|H| = (Nm+1)^\ell = 2^{N^{o(1)}}$.

We do simulations of the machine M for $E(\{p_m\})$ to get all the bits of all the evaluations of $p_m(\cdot)$ on S^ℓ. As p_m is multilinear with coefficient of at most $m^{e'}$ many bits, we can bound the bit size of any such evaluation by $O(m^{e'} \log(Nm))$. This means that the inputs $(1^m, a_1, \ldots, a_m, i, b)$ that we simulate M on, have bit size $O(m \log(Nm))$ (recall i is given in binary). For the string β we give the advice that M needs for all input lengths up to this maximum bit size, which is $O(m^2 \log^2(Nm)) = N^{o(1)}$ in total. Given such advice, the simulation for a single such input thus costs NTIME$[2^{O(m \log(Nm))}]$ = NTIME$[2^{N^{o(1)}}]$. To get all bits of an evaluation for a single element in H therefore takes at most NTIME$[O(m^{e'} \log(Nm)) \cdot 2^{N^{o(1)}}]$ = NTIME$[2^{N^{o(1)}}]$ with the same amount of advice. We conclude that we can construct the entire set H in NTIME$[2^{N^{o(1)}}]$ with $N^{o(1)}$ advice.

If during the process of obtaining all the bits we obtain a flag bit set to 0, we reject. This means that if on every path where we pass this check, we have obtained a hitting set, provided N is large enough. On the path where we pass this check, we continue to verify deterministically that $f(h) = 0$ for all $h \in H$. If yes, then we accept, else reject. By our previous remarks, for infinitely many N, this correctly decides whether $\Phi \equiv 0$.

Let us consider the cost of evaluation of Φ on elements of H. For $a \in S^\ell$ and subset S_j in the Nisan-Wigderson design, the bit size of $p_m(a_{|S_j})$ by $O(m^{e'} \log(Nm))$. By Corollary 8 this means that the absolute value of any gate of Φ for input $NW^{p_m}(a)$ is at most $2^{O(N^2 m^{2e'} \log^2(Nm))} = 2^{N^{O(1)}}$. Thus intermediate values can be represented by $poly(N)$ bits. We conclude that evaluation of Φ on a single element of the test set H cost time $poly(N)$. We can conclude the entire cost of our test algorithm is $NTIME[2^{N^{o(1)}}]$ with $N^{o(1)}$ advice, and that for infinitely many input lengths N the algorithm is correctly decides low-PIT. ◀

Due to space restriction the randomness-to-hardness direction of the proof of Theorem 6 has been ommited from this version of the paper. It will appear in the full version.

References

1 S. Aaronson and A. Wigderson. Algebrization: A new barrier in complexity theory. *Transactions on Computation Theory*, 1(1), 2009.

2 L. Babai, L. Fortnow, and C. Lund. Nondeterministic exponential time has two-prover interactive protocols. *Computational Complexity*, 1:3–40, 1991. Addendum in vol. 2 of same journal.

3 P. Bürgisser. *Completeness and Reduction in Algebraic Complexity Theory*. Springer Verlag, 2000.

4 R. DeMillo and R. Lipton. A probabilistic remark on algebraic program testing. *Inf. Proc. Lett.*, 7:193–195, 1978.

5 D. van Melkebeek J. Kinne and R. Shaltiel. Pseudorandom generators, typically correct derandomization, and circuit lower bounds. Technical Report TR10–129, Electronic Colloquium on Computational Complexity (ECCC), 2010.

6 M. Jansen. Extracting roots of arithmetic circuits by adapting numerical methods. In *Proc. 2nd Symp. on Innovations in Computer Science*, 2011.

7 V. Kabanets and R. Impagliazzo. Derandomizing polynomial identity testing means proving circuit lower bounds. *Computational Complexity*, 13(1–2):1–44, 2004.

8 P. Koiran. Shallow circuits with high powered inputs. In *Proc. 2nd Symp. on Innovations in Computer Science*, 2011.

9 P. Koiran and S. Perifel. Interpolation in Valiant's theory. *Computational Complexity*, 20(1):1–20, 2011.

10 N. Nisan and A. Wigderson. Hardness versus randomness. *J. Comp. Sys. Sci.*, 49:149–167, 1994.

11 R. Santhanam. Circuit lower bounds for Merlin–Arthur classes. *SIAM J. Comput.*, 39(3):1038–1061, 2009.

12 N. Saxena. Progress of polynomial identity testing. Technical Report ECCC TR09-101, Electronic Colloquium in Computational Complexity, 2009.

13 J.T. Schwartz. Fast probabilistic algorithms for polynomial identities. *J. Assn. Comp. Mach.*, 27:701–717, 1980.

14 V. Strassen. Vermeidung von divisionen. *Journal für die Reine und Angewandte Mathematik*, 264:182–202, 1973.

15 S. Toda. PP is as hard as the polynomial-time hierarchy. *SIAM J. Comput.*, 20:865–877, 1991.

16 L. Valiant. The complexity of computing the permanent. *Theor. Comp. Sci.*, 8:189–201, 1979.

17 R. Williams. Non-uniform ACC circuit lower bounds. In *Proceedings of 26th IEEE Conference on Computational Complexity*, page To appear, 2011.

18 R. Zippel. Probabilistic algorithms for sparse polynomials. In *Proceedings of the International Symposium on Symbolic and Algebraic Manipulation (EUROSAM '79)*, volume 72 of *Lect. Notes in Comp. Sci.*, pages 216–226. Springer Verlag, 1979.